An Evaluation of Computer Assisted Clinical Classification Algorithms

Christopher G. Chute
Yiming Yang
James Buntrock

Section of Medical Information Resources
Mayo Clinic/Foundation
Rochester, MN

The Mayo Clinic has a long tradition of indexing patient records in high resolution and volume. Several algorithms have been developed which promise to help human coders in the classification process. We evaluate variations on code browsers and free text indexing systems with respect to their speed and error rates in our production environment. The more sophisticated indexing systems save measurable time in the coding process, but suffer from incompleteness which requires a back-up system or human verification. Expert Network does the best job of rank ordering clinical text, potentially enabling the creation of thresholds for the pass through of computer coded data without human review.

INTRODUCTION

Interest in Health Care Reform has highlighted the need for well organized indicies to clinical information. Manual classification of diagnoses, procedures, and findings remains the standard method for creating these indicies, although "auto-coder" technology has been introduced to harness computer assistance to the task. Most computer assisted tools today attempt to navigate the user to a correct code using hierarchical menus. Few attempt to pattern match natural language entries from a health provider to a target coding system, such as ICD-9-CM or SNOMED-3.

The Mayo Foundation has maintained careful indicies to its "master sheet" summarization of findings and diagnoses upon dismissal since 1909 [1]. The clinical classifications have changed over time, but today involve a highly extended derivative of HICDA-2 [2]. numbering 29,762 discreet rubrics. The Section of Medical Information Resources at Mayo is operationally responsible for this coding, and expends over $1.4 million annually in the effort. The resolution and purpose of the master sheet entries has been incompatible with the reimbursement needs of business office coding, and therefore this specialized coding for research retrieval has been done in parallel.

We have developed several techniques for statistically based information retrieval, and have applied them to the free text phrase classification problem [3][4][5][6]. We have also developed a workstation application to assist our coding personnel in the classification process, so that we can code more clinical data with the same or fewer resources. In this paper, we outline our preliminary experience with alternative techniques for pattern matching free text diagnoses, and the process we are using to continuously improve our machine assisted classification tools.

METHODS

This evaluation was conducted on inpatient and outpatient summary diagnoses entered on Mayo Clinic's "Master Sheet;" these are typically 3-10 words of descriptive free-text. The Section of Medical Information Resources codes over 1.3 million diagnoses and findings from the paper based summary master sheet entries of the Mayo medical record annually. After more than a year of prototyping, X-terminal workstations were installed on the desks of all master sheet coders to improve speed and precision of coding. In anticipation of an on-line master sheet by 1995, coders type the master sheet text into a window of the coding application. This text is spell checked against a locally developed lexicon of 107,000 master sheet words and variants, using a proprietary fuzzy match algorithm (Proximity Scan P2 Library, Proximity Technology, Ft. Lauderdale, FL), however the human coders may accept terms not known by the lexicon.

For our evaluation, we employed five algorithms for matching the natural language text strings from the master sheet to the 29,762 codes in our locally extended version of HICDA-2; these include:

- Exact Browse - This method employs an exact string match on shared words and terms between the text string of the HICDA-2 codes and the master sheet entry, supplemented by browsing words entered by the coder.
- Fuzzy Browse - As exact browse, but string matching can be partial. String closeness is ranked by the Proximity Scan software.

0195-4210/94/$5.00

- Hashed Index - Master sheet words are normalized to match the lexical variants and spelling of the local lexicon, and then matched to the HICDA-2 coding system. Matches are ranked by the number of word strings that match the coding system entry terms.
- Least Squares Fit - A statistically based technique that learns associations between text phrases and their humanly assigned classification[4][5] . This technique is very computationally intensive, and is therefore solved in parts. For the 49,262 training set pairs used in this evaluation, 24 sub-set matrices were independently solved and later merged.
- Expert Network - A new approach that avoids the cubic computational complexity of Least Squares Fit, invoking a linear learning pattern.[6][7]. For this evaluation, 235,422 training pairs form the knowledge base. No problem sub-division was required, obviating a sub-set merge.

For each of these five techniques, approximately 1,000 master sheet entries were randomly chosen for coding. Trained personnel used clinical coding workstation software, specially developed for this evaluation, that invoked only one of the algorithms. A subsequent review by a supervisor edited the work, correcting oversights attributable to the software or human error. Supervisor judgment at this step included what constituted a codable master sheet clinical finding (as opposed to an administrative notation); this contributed to the variable number of verified items among the evaluated techniques in the final tally. The master sheet entries used to evaluate each coding system were not identical to save costs, but were drawn from the same source pool and are functionally equivalent.

The project software also tracked wall clock timing information, separating master sheet term typing time from coding time. Since electronic transfer of master sheet text is imminent at Mayo, we discounted the contribution of entry typing .

To provide comparable statistics, the three algorithms (not browsers) that returned a rank ordered list of matches were restricted to the top 40 matches found. Statistics compiled for each algorithm include the number of text strings coded, the resulting number of codes generated (allowing multiple codes per text), the percent of verified codes not included among the top 40 matches, average time to code in seconds, percent of initial codes corrected by the supervisor verification (codes with error), and the average rank order of the correct code among the 40 matches returned (based on last code in the event of multiple codes needed).

All processing was done on text in real time, with results ranked results (for the non-browsers) returned to the coder in under three seconds. We are striving for sub-second response time, but the human delay in review and confirmation of the best code in this computer assisted coding scenario overwhelms machine time contributions for now. Average rank statistics were computed after the correct answers were validated by the supervisor reviewer; the statistics were computed only for the 83% of texts which had a single correct code.

RESULTS

Table 1 shows the evaluation statistics of the five systems.

Number of Texts: each system uses a different testing set, and the sizes of these sets are similar but not exactly the same.

Number of Codes: the majority (83%) of our DXs had

Method	Number of Texts	Number of Codes	Candidate Codes Passed to Coders		Codes Not Found
Exact Browse	904	1112	unlimited		0.18%
Fuzzy Browse	1000	1224	unlimited		0.49%
Hashed Index	1197	1444	40		26%
Least Squares Fit	1025	1252	40		11%
Expert Network	1077	1340	40		16%
	Average Precision	Average Recall	Average Rank	Coding Error	Avg Time to Code
Exact Browse	–	–	–	5.1%	59 sec
Fuzzy Browse	–	–	–	5.0%	46 sec
Hashed Index	60%	72%	4.94	3.0%	24 sec
Least Squares Fit	45%	88%	4.63	8.8%	23 sec
Expert Network	83%	81%	1.63	2.5%	34 sec

Table 1. Performance of 5 computer assisted classification algorithms. See text for explanation of statistics.

exactly one correct code, the remaining DXs have 2-6 correct codes.

Candidate Codes Passed to Coders: the browsers allow a coder unlimited inquiry; the classifiers (Hashed Index, Least Squares Fit and Expert Network), on the other hand, return 40 top-ranking candidate categories for each text to the coder to choose.

Codes not Found: For the classifiers, the percentage of missed codes simply means the correct codes which are not included in the 40 top-ranking candidates. Browser figures reflect percent overlooked.

Average Time to Code: this includes system response, the time the coder checks through candidate categories returned by the system, and the coding decision. In case the correct code is not in the candidate list, the coder makes the decision based on knowledge.

Coding Error: the error rate of humans when using one of our systems to assist the coding. The error rate using manual methods is 7% on average.

Average Precision: defined as the ratio of the number of codes found and correct divided by the number of codes found. For each text, we computed a precision value at each position in the ranked candidate list where a correct code is found. If a text has more than one correct codes, then it can have more than one precision value; we average these values into a single measure for this text. The precisions of individual texts are further averaged for a global measure of a method. For the browsers, no such information was available to compute the precisions.

Average Recall: defined as the ratio of the number of codes found and correct divided by the number of correct codes. For the classifiers, we computed a recall for each text at the end (the 40th position) of the ranked candidate list, and then averaged the recalls of all texts for a global measure of a method. For the browsers, no such information was available.

Average Rank: the average rank of a correct code in the 40 element candidate list. It gives intuitive idea about how well a system does if all DXs have one correct code. In such a case, the best possible average rank is 1, and the smaller the average rank, the better the system. However, the average rank is problematic for DXs which have more than one correct code. For example, suppose the DXs have exactly two correct codes, then the best possible average rank is 1.5 (assuming the correct codes appear in the 1st and 2nd positions on the ranked list), but not 1 (assuming one correct code is found in the 1st position but the other correct code is simply missed)! Therefore, in this statistic, we only counted the DXs (83% of the total) which had only one code.

The table summarizes the statistics generated in our evaluation. Each technique was tested with a comparable number of input texts, and generated a similar number of resulting codes. The browser techniques had very few codes not found in the initial pass (not presented), by the nature of freely navigating browsers. The index and ranking algorithms were restricted to 40 possible matches in their returning window, which in this evaluation allowed between 11% to 26% correct codes to be overlooked.

The browser techniques took nearly twice as long, on average, to code a given text phrase. Expert Network took somewhat longer than the other indexing methods, due in part to its larger knowledge base of nearly one quarter million data pairs. Least Squares Fit was the fastest technique, but this does not account for the several hours of SPARC 10 time needed to compute the intervening matrices.

Least Squares Fit returned correct answers that, on average, were five or six lines from the top on the rank ordered list returned by the algorithm. Error rates for this system were also the highest (8.8%), perhaps due to the relative burying of correct answers further down the list. Despite these problems, Least Squares Fit tended to have the lowest rate of failure (11%) to include the correct answer among the restricted set of 40 potential matches returned for human review.

Expert Network tended to do the best at correctly ranking validated responses among the codes it did find. This appears to correlate with the error rate found in review that required correction (2.5%). It was intermediate in failure to return correct codes (16%). Time performance of Expert Network was better than the browsers, but worst among the indexing methods.

The browsers were slow and generated an intermediate number of errors. An average rank statistics has no meaning, since the browsers allowed coders to look among all 29,762 codes rather than a rank ordered list of 40 potential matches.

DISCUSSION

This evaluation shows that indexing algorithms can be faster and generate fewer errors than coding system browsers. Intelligent (fuzzy matching) browsers are faster than simple string browsers, and intelligent algorithms are faster still. For the time being however, a sufficient number of correct codes are

missed altogether that human review and confirmation is required to ensure high fidelity coding.

The fastest system, Least Squares Fit (LSF), also had the highest rate of human error and a precision significantly lower than Expert Network and Hashed Index. The low precision is probably due to an improper decomposition of the problem. That is, our current algorithm is not efficient enough to handle the large training sample used in this evaluation, so we split the training sample (about 50000 DXs) into subsets of 24 subdomains of the ICD-9-CM categories. We computed an LSF solution for each subset, and use these local solutions to estimate category scores in each subdomain. In principle, these local scores ("local evidences") should be used in combination with "global evidences", i.e. the likelihood estimates of a text as a concept under each subdomain. We have a method under development to integrate these two kinds of evidences [4]; however, for the time being, we only used local evidences in LSF for this evaluation. Such a temporary arrangement unavoidably introduced local biases in category ranking, and evidently caused the significant decrease of the precision (in our previous tests on smaller data sets where the training samples were not split, LSF had the precisions similar to Expert Network [6] [7]). The recall, on the other hand, is higher than the other systems.

The question is, how much does the precision (and recall) of a system effect the quality and cost of human coding in an interactive computer assisted coding environment? In Table 1, we observe a strong correlation between the low precision of LSF and the high rate of human errors; on the other hand, no such correlation is evident between precision and coding time. A low precision means that correct codes have relatively high ranks in the candidate list brought to the coder. That is, the coder has to check through many alternatives until a correct code is found. Since experienced coders are fast in checking through candidates, as long as the correct codes are included in the 40 top-ranking candidate list, their relative low ranks did not seem to slow down the coding speed by much. On the other hand, if a correct code is missed in the candidate list, it takes a much longer time for the coder to figure out the code. This means that the recall among the 40 top-ranking candidate list is probably more important than the precision from the view point of coding speed. LSF had the highest recall and also the fastest response time (about 1 second per query), all together it made the coding time by humans the shortest.

The cause of the high human errors may or may not be the low precision of LSF. It is possible that a coder would be confused by the low ranked alternatives in a candidate list. If this is true, then a precision enhancement would solve the problem. Our present work in sparse matrix algorithms and parallel computing, may significantly reduce the number of training subsets; by combining local and global evidencies in a "split-merge" method, we expect a substantial improvement in precision, and possibly in error rate. Another potential reason for human error would be the "scrambled" mixture of candidates, i.e. adjacent codes in the candidate list can be totally different concepts, and this may be very disorienting for coders. A solution for this problem is to improve the representation of the candidates, e.g. instead of giving a ranked list of codes, grouping the codes by concepts and laying out the grouped codes for coders to review.

Expert Network shows the precision and lowest error rate, approaching the point of permitting computers to indicate correct codes without human review. This evaluation did not exploit the information contained in the similarity matching scores which are integral to Least Squares Fit and Expert Network. Future work must establish if there are threshold values which will permit the confident acceptance of computer matches, without further human review or editing. Passing only 10-15% of such codes in this way would imply millions of dollars in savings to the health care industry in reduced coding costs.

Several limitations exist in this work. Most notably, the evaluations were not conducted on identical subsets of data. Our experience indicates that system performance is consistent over similar data types, somewhat attenuating this concern. Our alternative was to create a standard dataset with standard answers. However, the training effect of coding this dataset using the same coding personnel would have been large and confounded the evaluation of algorithms later in the sequence. While we could have used different persons, we reasoned that the variation in source material was smaller than the variation in coder consistency with these experimental techniques. We therefore opted to use different source material to enable us to use a consistent panel of human coders.

The generalizability of these findings is also not tested. We restricted this testing to our high volume production coding in the Section of Medical Information Resources, because that is where we targeted the development and early implementation of the system. However, the target coding space is an idiosyncratic adaptation of HICDA-2, which is much larger and more specific than ICD-9-CM, and architecturally unrelated (non-axial) to SNOMED. As we broaden the implementation of our coding algorithms in the Mayo environment, we will be able

to test how consistently these findings apply to alternative coding systems.

We have demonstrated that Computer Assisted Coding workstation tools save time in a production coding environment for a large, tabular coding system. Several algorithms make few errors, and Expert Network generates average rankings statistics that create substantial interest in threshold value research for complete automation of parts of the coding system.

ACKNOWLEDGMENTS

Supported in part by NIH grants LM05416, LM07041, and AR30582. We thank Geoffrey Atkin for computer support and Karen Elias for manuscript assistance. We specially thank Lorraine Fiksdal, Joan Wooner, and the medical indexing staff for their efforts and patience.

REFERENCES

[1] Kurland LT, Molgaard CA. The patient record in epidemiology. *Scientific American* 1981;245(4):54-63.

[2] HICDA-2, Hospital Adaptation of ICDA, 2nd Edition. Ann Arbor, MI: Commission on Professional and Hospital Activities, 1968.

[3] Chute CG, Yang Y. An evaluation of concept based Latent Semantic Indexing for Clinical Information Retrieval. *Proceedings of the 16th Annual Symposium on Computer Applications in Medical Care (SCAMC 92)* 1992;16:639-43.

4] Yang Y, Chute CG. A Linear Least Squares Fit Method for Terminology Mapping. *Proceedings of Fifteenth International Conference on Computational Linguistics (COLING 92)*, 1992;II:447-53.

[5] Yang Y, Chute CG. An application of least squares fit mapping to clinical classification. *Proceedings of the 16th Annual Symposium on Computer Applications in Medical Care (SCAMC 92)* 1992;16:460-4.

[6] Yang Y . Expert Network: Combining Word-based Matching and Human Experiences in Text Categorization and Retrieval. *Proc 17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR94), in press.

[7] Yang Y, Chute CG. An application of Expert Network to Clinical Classification and MEDLINE Indexing. *Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care* 1994, submitted.

[8] Yang Y, Chute CG. An application of least squares fit mapping to text information retrieval. *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 93)* 1993;281-90.